Massachusetts Institute of Technology
6.895: Advanced Distributed Algorithms
Professor Nancy Lynch

Lecture 4
February 22, 2006

Localization

These notes cover the first of two lectures given on the topic of localization. This lecture introduces the techniques and systems that have been developed, while the second lecture will cover the theoretical underpinnings of localization.

These notes are divided as follows. Section 1 is an introduction to localization, outlining the problem and introducing current techniques. Section 2 discusses various methods for ranging, the process of determining the physical distance between two transmitting nodes. Section 3 covers Cricket, a decentralized system that uses inexpensive beacons to broadcast location data within small spaces. Section 4 covers AHLoS, a system that uses a distributed algorithm to iteratively locate as many nodes as possible in an ad-hoc network. Section 5 covers mobile-assisted localization, a technique that uses a mobile user to add rigidity constraints to a Cricket localization network.

1 Introduction

1.1 The Localization Problem

For this discussion of localization we will make several assumptions:

- We will work in an ad-hoc setting, where some nodes may be mobile.
- The MAC layer achieves reliable communication between closely neighboring nodes.
- If necessary, we will assume that nodes have synchronized clocks.
- There may be nodes who start out knowing their own location, precisely or approximately. We call these nodes *anchors* (to prevent confusion with other types of *beacons*).

The problem of localization is for each node in the network to determine "reasonable" coordinates for itself. These coordinates may be absolute or relative to other nodes in the network. The goal of localization is for nodes to determine coordinates that deviate minimally from their actual physical location.

Why is localization a desirable goal? Location information may be used for a variety of applications:

- Routing point-to-point messages (in some networks, physical distance correlates highly with packet routing time)
- Broadcast (in a network broader than a single transmission range, we can ensure that all edge nodes have received a transmission)
- Aggregating data to central location (we know where the data is coming from / whom to wait for)
- Tracking targets (get location information from the nearest node)
- Implementing a virtual infrastructure (ensuring that all nodes are well integrated into the network)

1.2 GPS

GPS is a localization technology which works on a global scale. Unfortunately, current GPS technology has several drawbacks that make infeasible to use GPS to locate all the nodes in a network. First, GPS requires a direct line of sight to a satellite, rendering it useless for indoor networks. Second, GPS accuracy (10-20 feet) is less than we would desire for small networks. Third, GPS receivers are expensive, too expensive to use in a sensor network. Finally, GPS devices are large and require lots of power. For localization indoors, with many nodes, or with high precision, we want a technology that is deployable everywhere, accurate within centimeters, and low cost.

1.3 Steps and Techniques

Current localization solutions are subdivided as follows:

1. They start (possibly) with some anchors, which know their locations, exactly or approximately. They may learn their locations from GPS, or may be manually configured.
2. Then they determine inter-node distances for some of the nodes, usually nodes that can communicate in one hop. This part of the algorithm is known as *ranging.*
3. They may go through additional phases, determining more distances, between other pairs of nodes, or refining the existing set of measurements.
4. Once they have a stable set of distances, the nodes must compute coordinates.
   (a) If there are “enough” anchor nodes and enough edges with distances, it will be possible to assign unique coordinates to all the nodes.
   (b) If there are too few anchor nodes or too few distances, it might not be possible to assign unique locations—many different combinations of locations might be consistent with the given information. Then the algorithm can fail.
   (c) Even if there are no anchor nodes, if there are enough edges with distances, it will be possible to assign coordinates to all the nodes, in a way that is consistent among the nodes. That is, it may be possible to obtain a relative rather than an absolute coordinate system.
5. Once coordinates have been assigned, we can evaluate them as follows:
   For each edge having a measured distance (obtained from ranging), compare the measured distance with the distance between the locations of the endpoints that are assigned by the algorithm. This yields a measure of error for the assigned coordinates. Error can result from inaccuracies in the initial anchor coordinates or in the distance measurements. Errors can also be introduced during the later computation phases.
6. Various optimization techniques can be used to adjust the coordinate assignments to reduce the error.

2 Ranging

All the systems discussed in these notes rely on a method of finding the physical distance between two nodes in a network that are within communication range. This process is called ranging. There are two basic techniques used to perform ranging: received signal strength and signal propagation time.

2.1 Received Signal Strength

An inaccurate way to do ranging is to measure the signal strength of a message at the receiver. The receiver then uses knowledge of the sender's signal power (this might be contained within the message) to determine the power loss. Finally the receiver applies its known model for signal propagation behavior to convert the power loss to a distance, thus estimating how far away the sender is.

Unfortunately, this is not accurate. Radio signal propagation behavior is highly dependent on the environment (obstacles, metal,...), and highly variable. The Savvides paper describes experiments that tried to get good results this way, but they failed in every case except for an extremely idealized one: an empty football field, with nodes all at the same height. In most real-world ad-hoc networks, ranging by received signal strength is wildly inaccurate.

2.2 Signal Propagation Time

The preferred way of measuring distance is to measure the propagation time of a signal. Radio signals travel at the speed of light (essentially instantaneous arrival), so it is not plausible to measure this time. Other signals travel much slower, such as ultrasonic signals (US).

The most popular method of measuring distance using propogation time is called Time Difference of Arrival (TDoA). A sending node will transmit a radio (RF) and ultrasonic (US) signal *at the same time*. Because the radio signal arrives essentially instantaneously and the ultrasonic signal takes much longer, the receiver can measure the time difference between the arrivals, and thus deduce the traveled distance. Using this technique, it is possible to get an accuracy of around 2 cm for nodes 3 m apart. Old US transmitters have a typical range of 3-4 m, while newer ones typically reach 10 m.

One problem with ultrasound signal propagation is that it is subject to multipath effects, and to variations with changes in the environment. It is desirable to "recalibrate" TDoA measurements according to these variations. Savvides et al. give a way to perform this calibration, given enough redundancy in the distance data. The Cricket system also uses RF/US TDoA ranging.

3 Cricket

Priyantha, Chakraborty, Balakrishnan. The Cricket Location-Support System

3.1 Overview

Cricket is a location-support system for in-building, mobile, location-dependent applications. Such applications might involve active position mapping and the discovery of resources and devices in the proximity of a user. Cricket relies on information from static, preconfigured, ceiling-mounted beacons to relay positional information to listener devices. Beacons do not communicate with each other, or support any larger network infrastructure: they simply send out position information. The goal of Cricket is not to give listeners fine-grained position information, but rather to help listeners decide what space they are in, at the granularity of a 4x4 feet square.

The design criteria of Cricket are the following:

- user privacy - Cricket listeners can learn their location, but aren't necessarily tracked by the system
- decentralized administration - the owner of a space configures the beacons that identify it
- network heterogeneity - the Cricket system is decoupled from other connectivity services; it only supports localization
- low cost - Cricket beacons are small and cost less than U.S. $10
- room-sized granularity - the goal is to demarcate space boundaries within one to two square feet; using multiple beacons in a space accomplishes this goal

3.2 Ranging technique

Beacons use RF/US TDoA ranging, as discussed in section 2.2. This technique provides listeners an accurate depicting of which beacon they are closest to. More than one beacon can broadcast the same space identifier, together marking the edges of a space. A Cricket listener identifies with the space of the beacon it detects as closest.

What happens when multiple beacons transmit their RF and US at approximately the same time? It is possible that a listener might mismatch an RF message from beacon A with an ultrasound pulse from station B, thus miscalculating its current position. These effects are amplified by the multipath effects of ultrasound, causing signal copies to arrive after the direct-line signal. How can we prevent such interference?

3.2.1 Techniques for Coping with Interference

Collisions in TDoA packets cause two problems for localization. First, data is lost in the collisions. Cricket solves this problem by having beacons retransmit frequently (4 times per second) and within a random distribution (this prevents repeated collisions). Second, collisions might cause listeners to associate the wrong RF and US signals. This is harder to deal with.

One solution is for beacons to attach beacon identification tags to each transmission. This is actually what Cricket does for RF messages. This would be useful in the US pulse as well, but it is difficult to pulse an ultrasound emitter quickly enough to encode a station ID in the pulse. The emitter has to "ring down" much like a regular audio speaker would after playing a beat.

So we are still left with the problem of matching US signals to their RF messages. One method that Cricket uses is to extend the RF signal length to contain the ultrasound pulse. A listener matches an RF with the first US signal within the RF message. This means that delayed multipath US signals from the same beacon will be ignored. Also, if two beacons transmit with their RF messages overlapping, listeners will be able to detect a collision and ignore the surrounding US pulses.

There are still a few cases where incorrect corellations can be made. For example, if a late (multipath) US signal from beacon A overlaps with the ensuing RF signal from beacon B, a listener might correlate RF-B with US-A, thinking that it is very close to beacon B. These are anomalous cases, and can be filtered out by taking the mode delay time from a series of transmissions from each beacon.

3.3 Results

The results of the paper are quite conclusive. Even very near the boundaries between spaces, listeners encounter error rates of less than 30%, enough to disambiguate the listener's position between two beacons. Configuration of the system is also simple and decentralized, requiring only the setting of a space identifier on each beacon. To demonstrate the utility of Cricket, its developers have implemented a range of services, such as active mapping on floorplans and resource discovery. Researchers found that Cricket gave such accurate measurements that another paper will discuss how the technology can be used to do more than space identification. Section 5 describes an effort to get accurate location information from the Cricket system.

4 Savvides

Savvides, Han, Strivastava. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors

4.1 Overview

Savvides et al. describe the Ad-Hoc Localization System (AHLoS), an iterative way of discovering the absolute position of every node in a network. The researchers assume an ad-hoc network, in which anchors that know their own location at any given time form some percentage of the nodes. The focus is on two-dimensional localization, and the ranging method is TDoA.

4.2 Assigning locations

The key behind AHLoS is an iterative algorithm, by which positional information propagates from the anchors to their neighbors, and on into the rest of the network.

4.2.1 Atomic multilateration

Suppose we are trying to estimate the position of a particular "unknown" node 0 in 2-space, with some anchor neighbors. Let the unknown coordinates be (x_0, y_0). For each anchor node i for which we have a measured distance d_i from node 0, we can write the error as:

$$err_i = d_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

Here, the x_i and y_i are known. This gives us a system of equations, one per available anchor link, yielding an optimization problem—try to minimize the errors. We need at least 3 such equations: If there are only two equations, we can't get a unique solution, in general. We get two circles, which (may) intersect in two different locations. If known locations are exactly correct, and distance measurements from three non-colinear anchoring neighbors are exactly correct, then we can achieve error 0.

Now suppose there are four neighboring anchors. This is more information than is needed to get the unique solution for the position of node 0, meaning we can consider one more unknown and use the equations to get one additional piece of information. Namely, let s be the now-considered-unknown speed of propagation of the US signal. Then all the distance estimates can be rewritten as st_i, where t_i is the time taken by the US signal from i to 0. Now we set all the errors equal to 0, and get four equations, which allow us to solve for a unique (x_0, y_0) as well as s. In this way, a node can calculate its model for the speed of the US signal at the same time it is calculating its coordinates. So, with enough connectivity, we don't need to start everyone off with a known speed s, and we can calibrate s to counteract environmental effects (changes in temperature, obstacles, etc). Cool!

4.2.2 Iterative multilateration

Atomic multilateration discovers the previously unknown location of one node. Iterative multilateration is the idea that after the new node's location is found, we can use that node as an anchor itself and repeat. The heuristic for picking the next node to "discover" is to choose the unknown node with the largest number of anchor neighbors. This is where the paper is a little unclear as to whether this algorithm is truly distributed. The most-neighbors heuristic seems to imply that iterative multilateration does not happen in parallel. How, then, do the nodes know whom to locate next? Do the nodes gossip? Is there a leader election? To make this truly distributed we probably want parallel iterative multilateration, possibly sacrificing accuracy (nodes locate themselves earlier and less precisely than they may need to).

As is the case for any imprecise iterative calculation, iterative multilateration involves the accumulation of error. TDoA measurements are not exact, and since each node's position is defined with respect to previous nodes, errors can grow at each stage. In the worst case, ultrasound propagation knowledge is uniformly skewed in one direction, resulting in repeated over- or underestimations of distance. TThis type of error grows geometrically: a 90% measurement accuracy, used twice, leads to 81% accuracy in the second stage of multilateration. The technical term for this is "Geometric Dilution of Precision". Of course, if measurement error is randomly distributed, it is possible that the accumulated error is less drastic in practice: if round 1 overestimates, and round 2 underestimates, some of the measured error will cancel out.

4.2.3 Collaborative multilateration

The final stage of localization in AHLoS is collaborative multilateration. There are situations in which some of the individual nodes do not have enough anchor neighbors, but the entire collection does. How does this work? The anchors A and unknown nodes U form a graph in which every node U has at least 3 neighbors (else one node will never triangulate its location). *Participating nodes* in this graph are defined as nodes that are either anchors or unknows with three participating neighbors. We can now set up error equations just as before, between every participating node and anchor, *and* between every two participating nodes. Now there are $2|U|$ unknown values (the x and y of every unknown node) and if the graph contains "enough" independent equations, we can solve for all the nodes in U at once. "Enough" here is a tricky concept, because no node has access to all the graph constraints. The paper doesn't mention either what "enough" nodes means, saying only that applying collaborative multilateration on a wider scale will be the subject of future work. Even still, the paper shows that adding collaborative multilateration in a network where the number of anchor nodes is small substantially increased the number of resolved nodes.

We can give one simple example of collaborative multilateration. See figure 1 for a situation where two unknown nodes, with 2 anchoring neighbors each, can collaborate to determine both of their positions. Node 2 has two possible positions, determined by the intersection of circles from nodes 1 and 3. Similarly, nodes 5 and 6 constrain node 4 to two locations. When nodes 2 and 4 communicate, they can determine that their measured separation constrains both nodes to the intermediate positions.

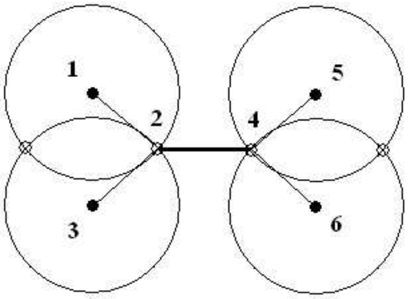

Figure 1: Good case for collaborative multilateration

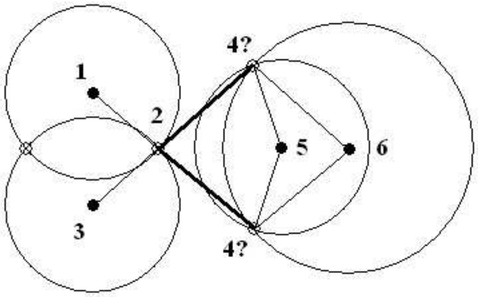

Figure 2: Bad case for collaborative multilateration

This kind of setup doesn't guarantee successful localization, however. Consider a similar scenario in figure 2, where now node 2 is equidistant from the two possible locations of node 4. This is a case in which the constraints do not give independent equations, and the location of node 4 remains uncertain.

4.3 Discussion

The placement of nodes and anchors in a network matters much when the localization algorithm depends on anchor nodes to disseminate position information. Ideally, a network would have a high degree of connectivity and beacons near everyone. Savvides et al. ran simulations of randomly positioned networks. They considered n nodes, placed on an L X L square field, with communication range R. For $L = 100$, $R = 10$, 200 nodes suffice to give good connectivity (high chance of more than 3 neighbors). The required fraction of anchors was also examined. Not surprisingly, fewer anchors are required as network density increases.

While the topic explored is interesting, and the results are encouraging, this paper is weak on several issues. First, the authors claim that running their algorithms in a distributed fashion is preferable. This avoids the cost of routing information to a central node. The iterative multilaterion algorithm, however, seems to assume some sort of leader election or network-wide knowledge. Second, the paper presents a distributed tree algorithm for collaborative multilateration, but doesn't justify or analyze this strategy with real graph theory. Third, the authors claim high accuracy for iterative multilateration, but admit that for large networks, improvement is still needed.

5 Mobile-Assisted Localization

Priyantha, Balakrishnan, Demaine, Teller. Mobile-Assisted Localization in Wireless Sensor Networks.

5.1 Overview

If nodes in a network are sparsely distributed, it may be difficult to produce a set of localization coordinates just from pairwise distances. The goal of Mobile-Assisted Localization (MAL) is to produce a set of coordinate assignments (2D or 3D) consistent with the pairwise distances, but based on additional information. MAL introduces temporary "virtual" nodes at strategic locations, using them to calculate distances between regular nodes that are otherwise out of each other's range. A virtual node is actually a handheld mobile device carried around by a human. Because there are no anchors in the MAL localization model, the resulting coordinate set is unique up to translation, rotation and reflection.

The paper also describes a method of obtaining relative coordinates called Anchor-Free Localization (AFL), and establishes some theoretical underpinnings in the form of graph rigidity theory.

5.2 Obtaining Enough Distance Measurements

In a real ad-hoc network, physical obstacles, limited ranging hardware and sparse node deployments might mean there are not enough distance measurements to constrain relative coordinates. That is, the graph might not be "globally rigid", or completely constrained. More edges reduces the depth of iterative multilateration, and hence the amount of error compounding. The use of mobile nodes is a great way to add edges to the network graph, because it's cheap and the administrator can use mobile nodes where the graph is sparse.

5.3 A little rigidity theory

Rigidity theory is the concerned with guaranteeing the uniqueness of structures in 2D or 3D space. This study has a large literature in varying disciplines such as structural engineering, molecular structures and pure math.

The following definitions describe complete (global) rigidity:
Definition: An *n-point formation* P in d-space consists of an assignment of coordinates to points $p_1, \ldots, p_n$ in d-space, and a set of *edges* (pairs of indices).
Definition: An n-point formation P in d-space is *globally rigid* provided that any other n-formation Q with the same edges and the same distances on those edges is the same as P, up to translation, rotation, and reflection.

There are other notions of rigidity. Local rigidity means that a structure cannot be locally deformed while preserving the distance constraints. Sometimes people use the term rigid to mean just locally rigid. In structural engineering, local rigidity implies that a structure cannot be deformed in 3-space.

A triangle is globally rigid in 2-space. A tetrahedron is globally rigid in 3-space. In fact, any fully-connected $(d + 1)$-point formation is globally rigid in d-space.

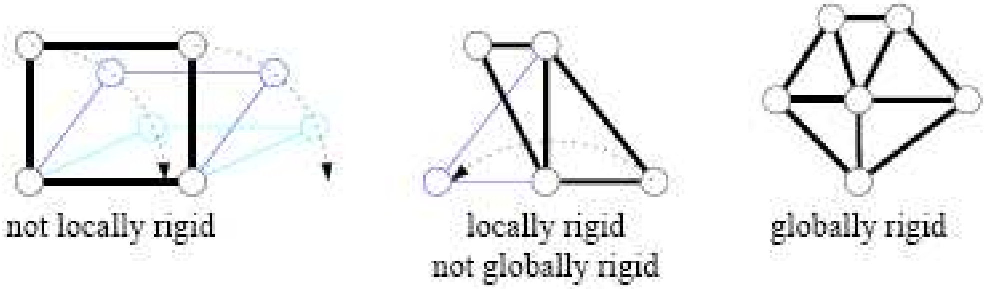

Figure 3: Examples of graphs that are not rigid (flexible as a bar-and-joint framework), rigid but not globally rigid (multiple embeddings), and globally rigid (one embedding up to rotation, translation, and reflection).

5.4 How MAL works

5.4.1 Subproblem: Determining distances between existing nodes

How exactly do you determine the distance between two nodes using external distance measurements? The MAL strategy uses a a set of geometric principles to determine just where virtual node distance measurements should be taken. In three dimensions, adding virtual nodes to a 2-node set is not straighforward, because each additional node adds three unknowns (its coordinates) but only 2 constraints (distances to the nodes). The trick is to take three distance measurements in a line, which contrains the measurements enough to determine the original nodes' physical separation. The paper describes at least three such tricks for contraining the node graph:

A point formation consisting of 5 coplanar points n_0, n_1, m_0, m_1, m_2, where m_0, m_1, m_2 are collinear, together with edges (n_i, m_j) for all i, j, is globally rigid.
A point formation consisting of 8 points $n_0, n_1, n_2, m_0, m_1, m_2, m_3, m_4, m_5$, where the n_i are not collinear and the m_j are all coplanar, and with edges (n_i, m_j) for all i, j, is globally rigid.
A point formation consisting of 11 points $n_0, n_1, n_2, n_3, m_0, m_1, m_2, m_3, m_4, m_5, m_6$, where no four points are coplanar, and with edges (n_i, m_j) for all i, j, is globally rigid.

5.4.2 Which distances to determine

The paper describes an algorithm for determining which distances to discover next. The general strategy is to start from a rigid structure and perform iterative multilateration with each new point connected to three previous points. The paper states an informal theorem, saying that their strategy will find a working mobile strategy, if one exists. They also claim measuring bounds linear in the number of given nodes.

5.5 AFL

The AFL (Anchor-Free Localization) algorithm is a localization alternative to Savvides-style iterative multilateration. AFL first assigns preliminary coordinates to all nodes based only on connectivity information of the graph, not on actualy distance information. This gives an inaccurate, collapsed version of the actual graph. AFL then “optimizes” for the minimal sum-of-squares error, taken over all edges. Each error value is the difference between the measured MAL edge distance and the distance computed from naive coordinate assignment. The goal is to transofrm the graph until error = 0, at which point a globally rigid graph is obtained that is unique up to translation, rotation and reflection.

5.6 Results

The experimental results show that MAL and AFL produce very accurate positional estimates, with errors of 10% or less. Mobile devices provide a convenient way of adding extra distance edges, that can produce a globally rigid structure. The separation between MAL measurements and AFL localization allows researchers to substitute other localization algorithms for localization and still get the benefits of MAL.